DELFT UNIVERSITY OF TECHNOLOGY

# REPORT 98-39

## Propagation of delays in public transport

Robert-Jan van Egmond

ISSN 0922-5641

Reports of the Faculty of Technical Mathematics and Informatics no. 98-39

Delft 1998

# Abstract

Synchronization in public transport can be modelled by means of Discrete Event Systems. Such a model can be used to determine how delays propagate, by means of simulation. In this paper an analytical approach to obtain the propagation of delays has been described.

# Keywords

Delays, Propagation, Discrete Event System, Synchronization

# Contents

**1 Summary** **2**

**2 Introduction** **2**

**3 Discrete event systems and (max,+) algebra** **3**
3.1 A train network example . . . . . . . . . . . . . . . . . . . . . . . . . . 3
3.2 The algebra . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4
3.3 Back to the example . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5
3.4 The critical circuit . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5

**4 Propagation of delays** **6**
4.1 Simulation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6
4.2 Analysis . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6
4.3 A single delay . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 9
4.4 Multiple delays . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 10
4.5 Buffer times . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 10
4.6 A performance measure . . . . . . . . . . . . . . . . . . . . . . . . . . . 12

**5 Conclusions** **12**

**Appendix A: Synchronization constraints** **14**

# 1 Summary

In public transport, modelling synchronization can be achieved by means of discrete event systems. In such models the travel times between stations are given and, with the aid of (max,+) algebra, the fastest possible timetable can be worked out (for instance the hour-pattern of the Dutch railway system takes at least 53.4 minutes). With this timetable we can then determine the propagation of delays. More precisely, we can calculate $M_{ij}$ which is the maximum delay at station $j$ that does not reach station $i$. These values can also be calculated when considering buffer times. Buffer times improve the reliability of the system and can also be used to create an appropriate cycle time (60 minutes instead of 53.4 minutes).
Two examples of the propagation of a delay are shown using the Dutch railway system as a model. In the first example the system has no buffer times and in the second example all buffer times equal 6.6 minutes (which will result in a cycle time of 60 minutes).
Finally, the usage of $M$ as a performance measure is discussed.

# 2 Introduction

Public transport is very disturbance sensitive due to weather conditions, the numbers of passengers getting in and out, traffic jams, accidents etc. which leads to many delays. As a consequence, travellers' trips become longer and moreover, because people miss their connections and have to wait at cold and boring transfer points, trips become less comfortable. Sometimes these connections can be maintained if trains or busses wait for each other. Such arrangements will be good for some travellers (those who keep their connection) but bad for others (those who are already waiting for the train or bus to depart). To keep passengers as satisfied as possible such decision rules have to be optimized.

Optimizing decision rules in public transport is not easy. One has to know for how many passengers the decision will be an advantage and for how many it will be a disadvantage, and also, how these advantages and disadvantages should be weighed up and whether or not a decision is at all feasible within the system. Furthermore, one not only needs knowledge of all these aspects at the time the decision is made, but also shortly afterwards because delays propagate in time.

In this paper we are concerned with calculating the propagation of delays. One way of calculating the propagation of delays is by doing simulations. However, this method is time-consuming and it does not give any further insight into the problem. We will therefore describe also an alternative calculation method, which is not time-consuming and which does give insight into the matter. This method is based on the theory of Discrete Event Systems (DES), a theory that has been used once before, to model the Dutch railway system [4]. In this paper we will use the same model of the Dutch railway system to give some examples.

# 3 Discrete event systems and (max,+) algebra

This section gives a brief introduction to the theory of discrete event systems and (max,+) algebra. A more detailed description can be found in [1]. First we start by giving a simple train network example.

## 3.1 A train network example

To explain the principles of discrete event systems, we shall consider an example of a small train network (cf. [3]). This network consists of two main stations denoted by marks 1 and 2, see Figure 1. At these main stations departing trains wait for

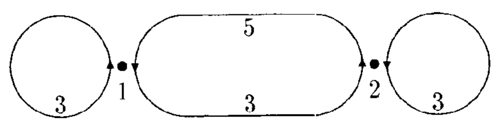



Figure 1: an example

arriving trains to give passengers the opportunity to change over. At the in-between stations trains leave immediately after passengers have got in and out. Since these in-between stations only affect the travelling times between the main stations they will be omitted from the model.
Four trains, denoted by the arrows in Figure 1, pass through this network. Figure 1 also shows the travel times for each route.
At a certain moment, say t=0, trains will depart from stations 1 and 2. At station 1 the first train will arrive at t=3 and the second train will arrive at t=5. Because these trains again wait for each other to depart, the next departure from station 1 will be at t=5. At station 2 both trains arrive at t=3 so these trains can therefore depart immediately, at t=3. This principle leads to the timetable shown in Table 1.

| departure | 1st | 2nd | 3rd | 4th | 5th | ... |
|---|---|---|---|---|---|---|
| station 1 | 0 | 5 | 8 | 13 | 16 | ... |
| station 2 | 0 | 3 | 8 | 11 | 16 | ... |

Table 1: timetable 1

A more regular timetable can be obtained if the trains at station 1 depart for the first time at t=1. This results in the timetable given in Table 2. According to this

| departure | 1st | 2nd | 3rd | 4th | 5th | ... |
|---|---|---|---|---|---|---|
| station 1 | 1 | 5 | 9 | 13 | 17 | ... |
| station 2 | 0 | 4 | 8 | 12 | 16 | ... |

Table 2: timetable 2

regular timetable trains leave from each station once in every 4 time units.

## 3.2 The algebra

The previous section described synchronization by means of an example. This section is concerned with the mathematical point of view. In the next section we will show how the equations found in this section can be applied to the example.

Given a network of $n$ nodes (i.e. stations) and connections, define:

$$\begin{array}{rcl} x_i(k) & : & \text{moment of the } k^{\text{th}} \text{ departure in node } i, \\ a_{ij} & : & \text{travelling time from node } j \text{ to node } i. \end{array}$$

Note that in (max,+) algebra going from $j$ to $i$ is denoted by $ij$, which is the other way round from usual notation.
The following equations describe the fact that events have to wait for each other:

$$\begin{aligned} x_1(k+1) &= \max\{x_1(k)+a_{11},\ldots,x_n(k)+a_{1n}\} \\ x_2(k+1) &= \max\{x_1(k)+a_{21},\ldots,x_n(k)+a_{2n}\} \\ &\vdots \\ x_n(k+1) &= \max\{x_1(k)+a_{n1},\ldots,x_n(k)+a_{nn}\} \end{aligned} \tag{1}$$

If no connection from node $j$ to node $i$ exists, we choose $a_{ij} = -\infty$. With the aid of (max,+) algebra we can write (1) in matrix form. This algebra differs from conventional algebra because of the following:

- besides reals we use the number $-\infty$; $\mathbb{R}_{\max} = \mathbb{R} \cup -\infty$,
- we replace addition by taking the maximum which will be denoted as $\oplus$,
- we replace multiplication by addition which will be denoted as $\otimes$.

Matrix multiplication in (max,+) algebra is defined as:

$$(A \otimes B)_{ij} = \bigoplus_{k=1}^{r} a_{ik} \otimes b_{kj} = \max_{k=1,\ldots,r}(A_{ik} + B_{kj})$$

Here $A$ and $B$ have sizes $m \times r$ respectively $r \times n$. Notice that this definition is similar to matrix multiplication in conventional algebra where $(AB)_{ij} = \sum_{k=1}^{r} a_{ik} \cdot b_{kj}$.

In (max,+) algebra we can write equations (1) as:

$$x(k+1) = A \otimes x(k) \tag{2}$$

The first set of events (the moments when the trains depart for the first time) will be denoted by $x(0)$. This vector, together with equation (2), determines the evolution of the following events. The main problem is to choose $x(0)$ so that a regular timetable appears, i.e. $x(k+1) = \lambda \otimes x(k)$ for some $\lambda$ and for all $k \in \mathbb{N}$. For this purpose we have to solve:

$$A \otimes v = \lambda \otimes v \tag{3}$$

As in conventional algebra $\lambda$ is called an eigenvalue and $v$ is called an eigenvector of matrix $A$.

## 3.3 Back to the example

Let us show how the mathematical framework of the previous section can be applied to the example network of Section 3.1. In the train network example train departures match the following equations:

$$
\begin{aligned}
x_1(k+1) &= \max\{x_1(k)+3, x_2(k)+5\} \\
x_2(k+1) &= \max\{x_1(k)+3, x_2(k)+3\}
\end{aligned}
$$

These equations can be written down in (max,+) algebra as:

$$x(k+1) = A \otimes x(k), \text{ where } A = \begin{bmatrix} 3 & 5 \\ 3 & 3 \end{bmatrix}$$

Suppose that each train departs for the first time at $t = 0$, which is denoted as $x(0) = [0, 0]'$, then, using equation (2) recursively, this will give:

$$x(1) = \begin{bmatrix} 5 \\ 3 \end{bmatrix}, x(2) = \begin{bmatrix} 8 \\ 8 \end{bmatrix}, x(3) = \begin{bmatrix} 13 \\ 11 \end{bmatrix}, \ldots$$

This corresponds to Table 1. A regular timetable would be established if trains at station 1 departed for the first time at t=1. Indeed $x(0) = [1, 0]'$ happens to be an eigenvector of $A$ with eigenvalue $\lambda = 4$:

$$\begin{bmatrix} 3 & 5 \\ 3 & 3 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 5 \\ 4 \end{bmatrix} = 4 \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

Following this starting vector $x(0)$ we obtain the same results as in Table 2. In general, it is not difficult to verify whether a given vector such as $[1, 0]'$ is an eigenvector or not. An eigenvector can be found by using the so-called power algorithm (cf. [2]).

## 3.4 The critical circuit

The eigenvalue of a system is directly related to the so-called critical circuit. A circuit is defined as a subset of nodes which is such that if we start in one of these nodes it will be possible to return to the same node by visiting the remaining nodes exactly once. The train network example consists of three circuits: a left, a middle and a right circuit (i.e. subsets {1}, {1,2} and {2}). The total travel time along a given circuit divided by the number of trains on the same circuit is called the circuit mean. According to a theorem in (max,+) algebra, the eigenvalue of a system equals the maximum circuit mean. In the example used here we have circuit means of respectively 3/1, (5+3)/2 and 3/1. The maximum circuit mean is thus 4. This is indeed the eigenvalue of the train network. Circuits that have a maximum circuit mean are called critical circuits.

# 4 Propagation of delays

In a railway system trains wait for each other. In what follows we assume that the connections remain the same, in spite of delays. Then the departure time of a train is the maximum of the timetable departure time and the arrival times of the preceding trains. As a consequence delays will propagate.
We will only consider delays that occur at nodes (stations). It should be noted that a delay at one node may stand for a delay of several trains. If we are dealing with a delay of a particular train, we first have to calculate the delay of the succeeding node reached by that train.
In this section we will assume that the network is strongly connected, i.e. it is possible to reach any node from any (other) node.

## 4.1 Simulation

Consider a timetable determined by an eigenvector $v$ of matrix $A$:

$$x(k) = \lambda^k \otimes v$$

Suppose at k=0 there are some delays $d(0)$ which result in a vector of disturbed moments of departure $\tilde{x}(0) = x(0) + d(0)$. Because all connections remain the same, the time-evolution of $\tilde{x}(0)$ can be found by applying matrix $A$ and timetable $x$:

$$\tilde{x}(k+1) = A\tilde{x}(k) \oplus x(k+1)$$

The difference between $\tilde{x}$ and $x$ determines the propagation of delay $d(0)$:

$$d(k) = \tilde{x}(k) - x(k)$$

This way of calculating the propagation of delays is thus merily a matter of simulation. Also when matrix $A$ is replaced by random matrices $A(k)$, i.e. the travel times are considered as stochastic variables, this way of simulation can be used.

## 4.2 Analysis

In this section we question which nodes will be disturbed by a particular delay. It is known that delays on the critical circuit never die out, since there is no slack between critical nodes (i.e. nodes on the critical circuit) to cath up. Moreover, every node in the (strongly connected) network will eventually be disturbed by any delay on the critical circuit (if the critical circuit is unique, as we will prove later). This has been illustrated in Figure 2, where the thick arrows indicate the critical circuit. On the other hand, a delay located off the critical circuit could die out before it has reached the critical circuit, but only if it is small enough, cf. Figure 3. Hence if a delay in a particular node is large enough, this delay or a part of it reaches the critical circuit and thereby reaches all other nodes. Considering two arbitrary nodes we question how large a delay in one of these nodes must be to affect the other node. This leads us to the following definition:

**Definition 1** *$M_{ij}$ is the maximum delay at node $j$ that does not reach node $i$.*

The values $M_{ij}$ can be obtained using the following lemma, but first we introduce some notations:

$$A^+ = A \oplus A^2 \oplus A^3 \oplus \cdots$$

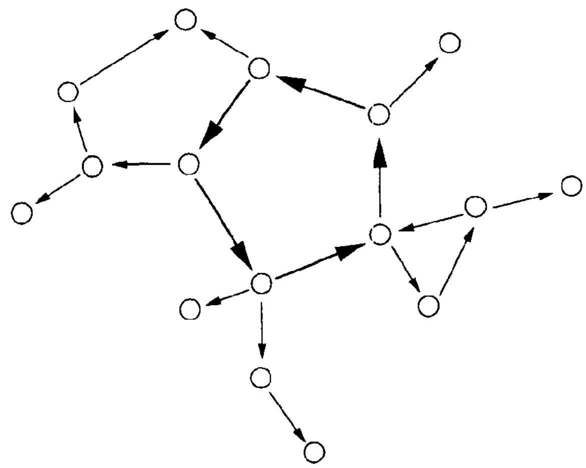


Figure 2: a delay on the critical circuit

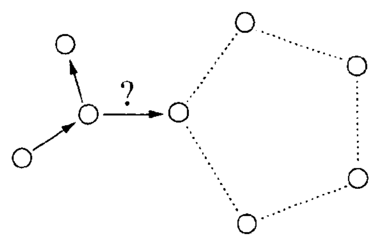



Figure 3: a delay off the critical circuit

The matrix $A^+$ is also refered as the *shortest path matrix.* Furthermore we will subtrackt the eigenvalue from each element of $A$ which will be notated by

$$(A_\lambda)_{ij} = a_{ij} - \lambda$$

Combining the above definitions gives

$$A_\lambda^+ = A_\lambda \oplus A_\lambda^2 \oplus A_\lambda^3 \oplus \cdots.$$

**Lemma 1** *Let $A$ be an irreducible matrix responding the travel times in a railway network with eigenvector $v$ and eigenvalue $\lambda$, then $M_{ij} = v_i - v_j - \left(A_\lambda^+\right)_{ij}$.*

*Proof.* Consider a path $\rho$ from node $j$ to node $i$ (such a path exists because the graph is assumed to be strongly connected). Renumber the nodes of this path as $1, ..., n$ (node $j$ becomes node 1 and node $i$ becomes node $n$).
Each pair of successive nodes on this path have slack, i.e. the departure time at the second node minus the arrival time of a train coming from the first node. Let us calculate the total slack on the whole path:

$$\begin{aligned}
\text{total slack of } \rho &= (v_2 + \lambda) - (v_1 + a_{21}) + \\
&\quad (v_3 + \lambda) - (v_2 + a_{32}) + \\
&\quad \cdots + \\
&\quad (v_n + \lambda) - (v_{n-1} + a_{nn-1}) \\
&= v_n - v_1 + \lambda - a_{21} + \lambda - a_{32} + \cdots + \lambda - a_{nn-1}
\end{aligned}$$

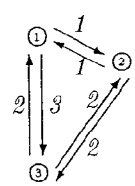
Let $P$ be the set of all possible paths from node $j$ to node $i$. The maximum delay in node $j$ that does not reach node $i$ equals the slack between node $j$ and $i$, minimized over $P$. Thus:

$$\begin{aligned}
M_{ij} &= \min_{\rho \in P} \left\{ v_i - v_j + \sum_{(k,l)\in\rho} (\lambda - a_{kl}) \right\} \\
&= v_i - v_j + \min_{\rho \in P} \left\{ \sum_{(k,l)\in\rho} (\lambda - a_{kl}) \right\} \\
&= v_i - v_j - \max_{\rho \in P} \left\{ \sum_{(k,l)\in\rho} (a_{kl} - \lambda) \right\} \\
&= v_i - v_j - \left(A_\lambda^+\right)_{ij}
\end{aligned}$$

The final equality is founded on the shortest path algorithm, cf. [1].

□

**Example 1** *Consider a network as drawn together with the corresponding $A$-matrix in Figure 4. One can easily validate that $v = [0, 0, \frac{1}{2}]'$ is an eigenvector with eigen-*




$$A = \begin{pmatrix} \varepsilon & 1 & 2 \\ 1 & \varepsilon & 2 \\ 3 & 2 & \varepsilon \end{pmatrix}$$

Figure 4: a three stations example

*value $\lambda = 2\frac{1}{2}$ and that the critical circuit is $\{1, 3\}$. Calculating $A_\lambda^+ = A_\lambda \oplus A_\lambda^2 \oplus A_\lambda^3 \oplus \cdots$ gives:*

$$A_\lambda^+ = \begin{pmatrix} 0 & -1 & -\frac{1}{2} \\ 0 & -1 & -\frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} & 0 \end{pmatrix}$$

*Furthermore, Lemma 1 gives:*

$$M = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

In Example 1 the columns of $M$ belonging to the nodes of the critical circuit turn out to be zero-columns. This means that any delay on the critical circuit eventually disturbs every node. The fact that $M_{ij} = 0$ whenever both $i$ and $j$ are nodes in the critical circuit is obvious: the critical circuit has no slack. The fact that $M_{ij} = 0$ also holds if only $j$ is a critical node is due to the fact that the critical circuit is unique as is claimed by the following lemma:

**Lemma 2** *If the critical circuit is unique and $j$ is a node of the critical circuit, then $M_{ij} = 0$ for every node $i$.*

To proof this lemma, we use the notion of triggering. We say that node $i$ is triggered by node $j$ if there exists a direct connection from $j$ to $i$ and the last train for which node $i$ has to wait is the train comming from node $j$, i.e. there is no slack between node $j$ and node $i$.

*Proof.* If both $i$ and $j$ belong to the critical circuit, $M_{ij} = 0$ is obvious. Let node $i$ be a noncritical node, i.e. a node off the critical circuit. Consider the set $S$ of all nodes that have a path to node $i$ without slack. Node $i$ is triggered by at least one node, which is thus in $S$. This node is then triggered by another node which is also in $S$, and so on. Because the number of nodes in the graph is finite, the set $S$ must contain a circuit. This circuit has no slack, so it is a critical circuit. Moreover, it is *the* critical circuit because the critical circuit was assumed to be unique. So, a path exists from the critical circuit to node $i$ without slack and thus $M_{ij} = 0$ whenever $j$ is a critical node.

□

Example 2 shows that this lemma does not hold if the critical circuit is not unique.

**Example 2** *Consider the following $A$-matrix of travelling times:*

$$A = \begin{pmatrix} \varepsilon & 2 & \varepsilon & \varepsilon \\ 2 & \varepsilon & \varepsilon & 1 \\ 1 & \varepsilon & \varepsilon & 2 \\ \varepsilon & \varepsilon & 2 & \varepsilon \end{pmatrix}$$

*This matrix has an eigenvector $v = [0, 0, 0, 0]'$ and eigenvalue $\lambda = 2$. There are two critical circuits: $\{1, 2\}$ and $\{3, 4\}$. Lemma 1 gives:*

$$M = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix}$$

*Although all nodes belong to a critical circuit, $M$ has non-zero elements. This shows that the uniqueness of the critical circuit in Lemma 2 is a necessary condition.*

## 4.3 A single delay

Let $D_j$ be a single delay in node $j$. What immediately follows from matrix $M$ is which nodes will be disturbed due to this delay and the maximum delay these nodes will get:

| | | |
|---|---|---|
| nodes disturbed by delay $D_j$ | : | $\{i \mid M_{ij} < D_j\}$ |
| maximum delay of node $i$ due to $D_j$ | : | $\max(D_j - M_{ij}, 0)$ |

In Example 3 we will illustrate this by using the Dutch railway network.

**Example 3** *Consider the $A$-matrix of the Dutch railway system, as built by Subiono in [4]. Appendix A, borrowed from [4], is a list of variables and departure constraints. Assume that an initial delay of 5 minutes occurs at node 2 (Utrecht). By means of simulation we obtain the propagation of this delay shown in Table 3. By the 8th time step all trains will be departing on time again. Table 3 also shows the non-zero elements of* $\max(5 - M_{i2}, 0)$. *These values are exactly the same as those obtained from simulation, although the information on the moments at which the delays occur is lost.*

| | | *simulation* | | | | | | | | $\max(D_j - M_{ij}, 0)$ |
|---|---|---|---|---|---|---|---|---|---|---|
| | | *time step* | | | | | | | | |
| | | *0* | *1* | *2* | *3* | *4* | *5* | *6* | *7* | |
| | *2* | *5* | | | | | | | | *0* |
| | *3* | | *5* | | | | | | | *5* |
| *d* | *4* | | | *5* | | | | | | *5* |
| *i* | *7* | | | | *4.4* | | | | | *4.4* |
| *s* | *13* | | *5* | | | | | | | *5* |
| *t* | *14* | | | *4.4* | | | | | | *4.4* |
| *u* | *79* | | | | *5* | | | | | *5* |
| *r* | *99* | | | | | | | *4.4* | | *4.4* |
| *b* | *101* | | | | *4.4* | | | | | *4.4* |
| *e* | *102* | | | | | *4.4* | | | | *4.4* |
| *d* | *103* | | | | | | *4.4* | | | *4.4* |
| | *119* | | | | | | | | *4.4* | *4.4* |
| *n* | *120* | | | | | | *4.4* | | | *4.4* |
| *o* | *121* | | | | | | | *4.4* | | *4.4* |
| *d* | *180* | | *5* | | | | | | | *5* |
| *e* | *195* | | | | *5* | | | | | *5* |
| *s* | *196* | | | | | *5* | | | | *5* |
| | *197* | | | *5* | | | | | | *5* |

Table 3: propagation of a 5 minutes delay at node 2 (Utrecht)

## 4.4 Multiple delays

When dealing with multiple initial delays, the maximum delay a node receives equals the maximum delay that node would receive from each single delay on its own. Let $D$ be a vector of initial delays. The maximum delay that node $i$ receives equals:

$$\max\left(\max_j\{D_j - M_{ij}\}, 0\right).$$

## 4.5 Buffer times

In order to increase the reliability of the railway system, buffer times are added to catch up with delays. Let $B$ be the matrix of buffer times, i.e. $B_{ij}$ is the buffer time of a train going from $j$ to $i$. A new timetable can be produced according to the new model $\tilde{A}$:

$$\tilde{A} = A + B$$

here, $+$ means the conventional addition of matrices. The new timetable and cycle time follows from the eigenvector and eigenvalue of $\tilde{A}$:

$$\tilde{A} \otimes \tilde{v} = \tilde{\lambda} \otimes \tilde{v}$$

In practice, we want $\tilde{\lambda}$ to be a round number (60 minutes in the Dutch railway system). This can be achieved by choosing $B$ appropriate, for instance by choosing $B_{ij} = \tilde{\lambda} - \lambda$ for each pair $i$ and $j$. It is easy to verify that when all buffertimes are equal, $\tilde{v}$ equals $v$.

Again, we want to know what is the maximum delay in node $j$ that does not reach node $i$. In the new model the slack between two successive nodes equals $(\tilde{v}_2 + \tilde{\lambda}) - (\tilde{v}_1 + a_{21})$, so we have:

$$\tilde{M}_{ij} = \tilde{v}_i - \tilde{v}_j - \left(A_{\tilde{\lambda}}^{+}\right)_{ij} \tag{4}$$

Notice that the lacking of the tilde above $A$ in (4) is not a misprint. On the contrary, it is essential that the buffertimes disappear in the synchronization constraints when calculating the propagation of delays.

**Example 4** *Consider the network given in Example 1 and add buffer times of* $\frac{1}{2}$ *to all the travel times:*

$$\tilde{A} = A \otimes \frac{1}{2} = \begin{pmatrix} \varepsilon & 1\frac{1}{2} & 2\frac{1}{2} \\ 1\frac{1}{2} & \varepsilon & 2\frac{1}{2} \\ 3\frac{1}{2} & 2\frac{1}{2} & \varepsilon \end{pmatrix}$$

*Then* $\tilde{\lambda} = 3$, $\tilde{v} = v = [0, 0, \frac{1}{2}]'$ *and, according to (4),*

$$\tilde{M} = \begin{pmatrix} 1 & 2 & \frac{1}{2} \\ 1 & 2 & \frac{1}{2} \\ \frac{1}{2} & 1\frac{1}{2} & 1 \end{pmatrix}$$

**Example 5** *Consider again the Dutch railway system, as in Example 3. The cycle time of the model without buffer times equals 53.4 minutes (cf. [4]). However the Dutch railway company has a schedule on an hourly basis and therefore uses a cycle time of 60 minutes. We therefore modify the model of Subiono by adding 6.6 minutes buffer time to all the travel times. An initial delay of 5 minutes at node 2 will immediately be absorbed by the 6.6 minutes buffer time. Table 4 gives*

| | | *simulation* | | | | | $\max(D_j - M_{ij}, 0)$ |
|---|---|---|---|---|---|---|---|
| | | *time step* | | | | | |
| | | *0* | *1* | *2* | *3* | *4* | |
| | *2* | *30* | | | | | *0* |
| *d* | *3* | | *23.4* | | | | *23.4* |
| *i* | *4* | | | *16.8* | | | *16.8* |
| *s* | *5* | | | | *0.2* | | *0.2* |
| *t* | *7* | | | | *9.6* | | *9.6* |
| *u* | *9* | | *15.2* | | | | *15.2* |
| *r* | *10* | | | *6.2* | | | *6.2* |
| *b* | *13* | | *23.4* | | | | *23.4* |
| *e* | *14* | | | *16.2* | | | *16.2* |
| *d* | *79* | | | | *10.2* | | *10.2* |
| | *101* | | | | *9.6* | | *9.6* |
| *n* | *102* | | | | | *3* | *3* |
| *o* | *180* | | *23.4* | | | | *23.4* |
| *d* | *195* | | | *5.2* | *10.2* | | *10.2* |
| *e* | *196* | | | | | *3.6* | *3.6* |
| *s* | *197* | | | *16.8* | | | *16.8* |

Table 4: propagation of 30 minutes delay at node 2 (Utrecht) with buffer times of 6.6 minutes

*the propagation of 30 minutes at node 2. This delay dies out after 5 time steps have been taken. Again, the results obtained from simulation and from using the* $M$*-matrix are shown.*

## 4.6 A performance measure

Matrix $M$ measures the propagation of delays and could be used as a performance measure. In fact, matrix $M$ represents the trade-off between robustness and synchronization. If $M$ is low, the degree of propagation is high, which will result in poor robustness but good synchronization. If $M$ is high, then the degree of propagation is low, robustness will be good and synchronization will be poor (cf. Table 5).

| | robustness | synchronization |
|---|---|---|
| high $M$ | good | bad |
| low $M$ | bad | good |

Table 5: matrix $M$ as a performance measure

For the purpose of performance measurement an ordering of matrices $M$ is required. One way of ordering would be by considering the mean of $M_{ij}$. This ordering has the advantage of being independent of the eigenvector $v$, since:

$$\sum_{ij} M_{ij} = \sum_{ij} \left( v_i - v_j - (A_\lambda^+)_{ij} \right) = -\sum_{ij} (A_\lambda^+)_{ij}$$

A disadvantage of the mean is its sensitivity to peaks. Peaks of $M_{ij}$ will occur especially on connections between minor stations that lie far away form each other. Since these connections are less important but seriously affect the ordering, a weighted mean should be considered.

# 5 Conclusions

In this paper we consider two means of calculating the propagation of delays; by simulations and by the matrix $M$.

A disadvantage of using matrix $M$ is that it does not give all the information which can be obtained from simulation. The information on the moments at which delays occur will be lost and if several delays occur at one node, only the largest one will be given.

The advantages of using $M$ instead of simulation are, firstly that $M$ can be calculated in advance. Then the propagation immediately follows when initial delays are given. By contrast, simulations must be done over and over again whenever initial delays are given. Secondly, using matrix $M$ gives more insight into the matter than using simulations; it gives the slack between each pair of nodes. Finally, matrix $M$ could be used as a performance measure. It represents the trade-off between robustness and synchronization.


## Acknowledgement

This publication is the result of the research programme Seamless Multimodal Mobility, carried out within the Netherlands TRAIL Research School for Transport, Infrastructure and Logistics, and financed by the Delft University of Technology.

# References


[1] F Baccelli, G Cohen, GJ Olsder, JP Quadrat *Synchronization and linearity (An algebra for discrete event systems)* (1992) John Wiley & Sons, Chichester

[2] JG Braker *Algorithms and applications in timed discrete event systems* (1993) Ph.D. Thesis, Delft University of Technology

[3] GJ Olsder *Timed Discrete Event Systems and Time Tables* (1996) ECMI, Progress in Industrial Mathematics, pp. 40-50, Wiley

[4] GJ Olsder, Subiono *On large scale max-plus algebra models in railway systems* (1997) Internal Report, Faculty of Information Technology and Systems, Delft University of Technology

# Appendix A: Synchronization constraints

This appendix is borrowed from [4].

| line | var. | departure from | destination | departure constraints |
|---|---|---|---|---|
| 1 | $x_1$ | Den Haag CS | Enschede | $x_6 + 38, x_{67} + 51, x_{108} + 12, x_{141} + 20$ |
| 1 | $x_2$ | Utrecht | Enschede | $x_1 + 40, x_6 + 72, x_{158} + 66, x_{165} + 59, x_{60} + 51$ |
| 1 | $x_3$ | Deventer | Enschede | $x_2 + 50$ |
| 1 | $x_4$ | Enschede | Den Haag CS, Rotterdam CS | $x_3 + 41, x_{195} + 10$ |
| 1 | $x_5$ | Deventer | Den Haag CS, Rotterdam CS | $x_4 + 41, x_{185} + 51$ |
| 1 | $x_6$ | Utrecht | Den Haag CS, Rotterdam CS | $x_5 + 53, x_8 + 39, x_7 + 40, x_{31} + 61, x_{152} + 36$ |
| 2 | $x_7$ | Den Haag CS | Leeuwarden, Groningen | $x_{14} + 38, x_{105} + 12$ |
| 2 | $x_8$ | Rotterdam CS | Leeuwarden, Groningen | $x_{14} + 36, x_{56} + 37, x_{58} + 42, x_{67} + 31, x_{96} + 45$ |
| 2 | $x_9$ | Amersfoort | Leeuwarden, Groningen | $x_7 + 55, x_8 + 54, x_2 + 15, x_5 + 37, x_{15} + 30, x_{16} + 36, x_{145} + 13$ |
| 2 | $x_{10}$ | Zwolle | Leeuwarden, Groningen | $x_9 + 35, x_{79} + 20, x_{190} + 23$ |
| 2 | $x_{11}$ | Leeuwarden | Den Haag CS, Rotterdam CS | $x_{10} + 54, x_{205} + 37, x_{209} + 51$ |
| 2 | $x_{12}$ | Groningen | Den Haag CS, Rotterdam CS | $x_{10} + 58, x_{201} + 41, x_{203} + 49$ |
| 2 | $x_{13}$ | Amersfoort | Den Haag CS, Rotterdam CS | $x_{12} + 93, x_{11} + 90, x_{15} + 30, x_{16} + 36, x_2 + 60, x_{176} + 34$ |
| 2 | $x_{14}$ | Utrecht | Den Haag CS, Rotterdam CS | $x_{13} + 16, x_{15} + 30, x_{16} + 36, x_{33} + 30$ |
| 3 | $x_{15}$ | Amsterdam CS | Enschede | $x_{21} + 30, x_{47} + 28, x_{61} + 15, x_{84} + 16$ |
| 3 | $x_{16}$ | Schiphol | Enschede | $x_{21} + 35, x_{61} + 29, x_{60} + 57, x_{104} + 19, x_{139} + 84, x_{137} + 9$ |
| 3 | $x_{17}$ | Amersfoort | Enschede | $x_{15} + 30, x_{16} + 36, x_7 + 55, x_8 + 54$ |
| 3 | $x_{18}$ | Hengelo | Enschede | $x_{17} + 68, x_{181} + 61$ |
| 3 | $x_{19}$ | Enschede | Amsterdam CS, Schiphol | $x_{18} + 12$ |
| 3 | $x_{20}$ | Almelo | Amsterdam CS, Schiphol | $x_{19} + 17$ |
| 3 | $x_{21}$ | Amersfoort | Amsterdam CS, Schiphol | $x_{20} + 60, x_{12} + 93, x_{11} + 90, x_7 + 55, x_8 + 54$ |
| 4 | $x_{22}$ | Amsterdam CS | Leeuwarden, Groningen | $x_{29} + 30, x_{31} + 33, x_{40} + 28, x_{47} + 62, x_{119} + 38$ |
| 4 | $x_{23}$ | Schiphol | Leeuwarden, Groningen | $x_{29} + 35, x_{57} + 26, x_{52} + 14, x_{107} + 19, x_{151} + 4$ |
| 4 | $x_{24}$ | Zwolle | Leeuwarden, Groningen | $x_{22} + 66, x_{23} + 72, x_{20} + 96, x_{73} + 50, x_{143} + 49$ |
| 4 | $x_{25}$ | Meppel | Leeuwarden, Groningen | $x_{24} + 15$ |
| 4 | $x_{26}$ | Leeuwarden | Amsterdam CS, Schiphol | $x_{25} + 46, x_{192} + 52, x_{194} + 48$ |
| 4 | $x_{27}$ | Groningen | Amsterdam CS, Schiphol | $x_{25} + 50, x_{191} + 53, x_{193} + 50, x_{199} + 44$ |
| 4 | $x_{28}$ | Zwolle | Amsterdam CS, Schiphol | $x_{26} + 64, x_{27} + 66, x_{73} + 50$ |
| 4 | $x_{29}$ | Amersfoort | Amsterdam CS, Schiphol | $x_{28} + 36, x_5 + 37$ |
| 5 | $x_{30}$ | Den Helder | Nijmegen | $x_{35} + 30$ |
| 5 | $x_{31}$ | Heerhugowaard | Nijmegen | $x_{30} + 36$ |
| 5 | $x_{32}$ | Utrecht | Nijmegen | $x_{31} + 61, x_{29} + 58, x_{35} + 40, x_{61} + 33, x_{151} + 35$ |
| 5 | $x_{33}$ | Ede-Wageningen | Den Helder | $x_{32} + 67, x_{112} + 49, x_{175} + 34$ |
| 5 | $x_{34}$ | Amsterdam CS | Den Helder | $x_{33} + 51, x_{21} + 30, x_{84} + 16, x_{151} + 17$ |
| 5 | $x_{35}$ | Heerhugowaard | Den Helder | $x_{34} + 39$ |
| 6 | $x_{36}$ | Den Helder | Arnhem | $x_{41} + 69, x_{124} + 53$ |
| 6 | $x_{37}$ | Amsterdam CS | Arnhem | $x_{36} + 69, x_{21} + 30, x_{57} + 41, x_{118} + 13, x_{144} + 12$ |
| 6 | $x_{38}$ | Ede-Wageningen | Arnhem | $x_{37} + 50, x_5 + 75, x_{138} + 53, x_{175} + 34$ |
| 6 | $x_{39}$ | Arnhem | Den Helder | $x_{38} + 12, x_{179} + 68$ |
| 6 | $x_{40}$ | Utrecht | Den Helder | $x_{39} + 33, x_5 + 53, x_{152} + 36$ |
| 6 | $x_{41}$ | Amsterdam CS | Den Helder | $x_{40} + 28, x_{88} + 35, x_{133} + 52, x_{135} + 65, x_{138} + 25$ |
| 7 | $x_{42}$ | Amsterdam CS | Maastricht | $x_{47} + 62, x_{21} + 30, x_{88} + 36, x_{95} + 45$ |
| 7 | $x_{43}$ | 's-Hertogenbosch | Maastricht | $x_{42} + 56, x_{148} + 44$ |
| 7 | $x_{44}$ | Sittard | Maastricht | $x_{43} + 70, x_{90} + 71, x_{111} + 36$ |
| 7 | $x_{45}$ | Sittard | Haarlem | $x_{44} + 31$ |
| 7 | $x_{46}$ | Eindhoven | Haarlem | $x_{45} + 46, x_{91} + 42$ |
| 7 | $x_{47}$ | Utrecht | Haarlem | $x_{46} + 50, x_{152} + 36, x_{171} + 39$ |
| 8 | $x_{48}$ | Haarlem | Eindhoven | $x_{51} + 17, x_{83} + 46$ |
| 8 | $x_{49}$ | Utrecht | Eindhoven | $x_{48} + 45, x_{163} + 12$ |
| 8 | $x_{50}$ | 's-Hertogenbosch | Haarlem | $x_{49} + 71$ |
| 8 | $x_{51}$ | Amsterdam CS | Haarlem | $x_{50} + 57, x_{21} + 30, x_{148} + 44, x_{160} + 40, x_{168} + 50$ |
| 9 | $x_{52}$ | Amsterdam CS | Vlisingen | $x_{57} + 41, x_{150} + 38$ |
| 9 | $x_{53}$ | Den Haag HS | Vlisingen | $x_{52} + 40, x_{95} + 59, x_{163} + 36$ |
| 9 | $x_{54}$ | Roosendaal | Vlisingen | $x_{53} + 52$ |
| 9 | $x_{55}$ | Vlisingen | Amsterdam CS | $x_{54} + 52$ |
| 9 | $x_{56}$ | Roosendaal | Amsterdam CS | $x_{55} + 51$ |
| 9 | $x_{57}$ | Den Haag HS | Amsterdam CS | $x_{56} + 53, x_6 + 52$ |
| 10 | $x_{58}$ | Schiphol | Roosendaal | $x_{61} + 29, x_{21} + 35$ |
| 10 | $x_{59}$ | Dordrecht | Roosendaal | $x_{58} + 57, x_{67} + 85, x_{172} + 46$ |
| 10 | $x_{60}$ | Dordrecht | Amsterdam CS | $x_{59} + 44, x_{76} + 45, x_{172} + 46$ |
| 10 | $x_{61}$ | Schiphol | Amsterdam CS | $x_{60} + 57, x_{21} + 35, x_{93} + 58, x_{164} + 42$ |
| 11 | $x_{62}$ | Den Haag HS | Heerlen | $x_{67} + 54, x_{52} + 40x, _{135} + 58$ |
| 11 | $x_{63}$ | Breda | Heerlen | $x_{62} + 48, x_{53} + 48, x_{76} + 41$ |
| 11 | $x_{64}$ | Eindhoven | Heerlen | $x_{63} + 36, x_{167} + 30$ |
| 11 | $x_{65}$ | Heerlen | Den Haag CS | $x_{64} + 66$ |
| 11 | $x_{66}$ | Roermond | Den Haag CS | $x_{65} + 32, x_{111} + 20$ |
| 11 | $x_{67}$ | Breda | Den Haag CS | $x_{66} + 71$ |
| 12 | $x_{68}$ | Zwolle | Roosendaal | $x_{73} + 96, x_{79} + 107, x_{234} + 17$ |
| 12 | $x_{69}$ | Arnhem | Roosendaal | $x_{68} + 56, x_{197} + 60, x_{23} + 72$ |
| 12 | $x_{70}$ | Tilburg | Roosendaal | $x_{69} + 59, x_{42} + 71$ |
| 12 | $x_{71}$ | Roosendaal | Zwolle | $x_{70} + 42, x_{53} + 52, x_{55} + 51$ |
| 12 | $x_{72}$ | 's-Hertogenbosch | Zwolle | $x_{71} + 50$ |
| 12 | $x_{73}$ | Arnhem | Zwolle | $x_{72} + 43$ |
| 13 | $x_{74}$ | Deventer | Roosendaal | $x_{79} + 39, x_{187} + 67, x_{190} + 42$ |
| 13 | $x_{75}$ | Nijmegen | Roosendaal | $x_{74} + 48, x_{112} + 27$ |
| 13 | $x_{76}$ | Breda | Roosendaal | $x_{75} + 66, x_{46} + 57, x_{66} + 71$ |
| 13 | $x_{77}$ | Breda | Zwolle | $x_{76} + 41, x_{62} + 48$ |
| 13 | $x_{78}$ | 's-Hertogenbosch | Zwolle | $x_{77} + 33$ |
| 13 | $x_{79}$ | Deventer | Zwolle | $x_{78} + 78, x_{197} + 52$ |
| 14 | $x_{80}$ | Amsterdam CS | Breda | $x_{84} + 16, x_{36} + 69, x_{21} + 30, x_{146} + 15$ |
| 14 | $x_{81}$ | Rotterdam CS | Breda | $x_{80} + 61, x_{52} + 63, x_{158} + 23$ |
| 14 | $x_{82}$ | Breda | Amsterdam CS | $x_{81} + 46, x_{75} + 66, x_{173} + 73$ |
| 14 | $x_{83}$ | Rotterdam CS | Amsterdam CS | $x_{82} + 47$ |
| 14 | $x_{84}$ | Haarlem | Amsterdam CS | $x_{83} + 46, x_{97} + 45$ |
| 15 | $x_{85}$ | Amsterdam CS CS | Dordrecht | $x_{88} + 35, x_{47} + 28, x_{31} + 33, x_{115} + 39x_{153} + 15$ |
| 15 | $x_{86}$ | Leiden | Dordrecht | $x_{85} + 34$ |
| 15 | $x_{87}$ | Dordrecht | Amsterdam CS | $x_{86} + 50$ |
| 15 | $x_{88}$ | Leiden | Amsterdam CS | $x_{87} + 52$ |
| 16 | $x_{89}$ | Dordrecht | Venlo | $x_{93} + 34$ |
| 16 | $x_{90}$ | Tilburg | Venlo | $x_{89} + 35$ |
| 16 | $x_{91}$ | Venlo | Rotterdam CS | $x_{90} + 64, x_{111} + 41$ |
| 16 | $x_{92}$ | Eindhoven | Rotterdam CS | $x_{91} + 42, x_{45} + 46$ |
| 16 | $x_{93}$ | Dordrecht | Rotterdam CS | $x_{92} + 57$ |
| 17 | $x_{94}$ | Hoorn | Rotterdam CS | $x_{98} + 50, x_{30} + 44, x_{114} + 27$ |
| 17 | $x_{95}$ | Alkmaar | Rotterdam CS | $x_{94} + 25, x_{30} + 36, x_{121} + 15$ |
| 17 | $x_{96}$ | Leiden | Rotterdam CS | $x_{95} + 49, x_{52} + 30, x_{119} + 44$ |
| 17 | $x_{97}$ | Rotterdam CS | Hoorn | $x_{96} + 45, x_6 + 36, x_{86} + 35$ |
| 17 | $x_{98}$ | Haarlem | Hoorn | $x_{97} + 66, x_{60} + 63$ |
| 18 | $x_{99}$ | Heerhugowaard | Rotterdam CS | $x_{103} + 66$ |
| 18 | $x_{100}$ | Leiden | Rotterdam CS | $x_{99} + 59, x_{58} + 16, x_{161} + 13$ |
| 18 | $x_{101}$ | Rotterdam CS | Hoorn | $x_{100} + 45, x_{14} + 36$ |
| 18 | $x_{102}$ | Den Haag HS | Hoorn | $x_{101} + 25$ |
| 18 | $x_{103}$ | Haarlem | Hoorn | $x_{102} + 41$ |

| line | var. | departure from | destination | departure constraints |
|---|---|---|---|---|
| 19 | $x_{104}$ | Amsterdam CS | Den Haag CS | $x_{106} + 37, x_{21} + 30$ |
| 19 | $x_{105}$ | Leiden | Den Haag CS | $x_{104} + 37, x_{80} + 34, x_{164} + 26$ |
| 19 | $x_{106}$ | Leiden | Amsterdam CS | $x_{105} + 25, x_{87} + 52$ |
| 20 | $x_{107}$ | Amsterdam CS | Den Haag CS | $x_{109} + 19, x_{29} + 30, x_{153} + 15$ |
| 20 | $x_{108}$ | Leiden | Den Haag CS | $x_{107} + 37, x_{85} + 34, x_{150} + 22$ |
| 20 | $x_{109}$ | Schiphol | Amsterdam CS | $x_{108} + 43$ |
| 21 | $x_{110}$ | Venray | Roermond | $x_{112} + 54, x_{112} + 41$ |
| 21 | $x_{111}$ | Venlo | Roermond | $x_{110} + 16, x_{90} + 16$ |
| 21 | $x_{112}$ | Venray | Nijmegen | $x_{111} + 56, x_{64} + 68, x_{45} + 50$ |
| 22 | $x_{113}$ | Amsterdam CS | Enkhuizen | $x_{115} + 39, x_{29} + 30, x_{109} + 19, x_{157} + 81$ |
| 22 | $x_{114}$ | Zaandam | Enkhuizen | $x_{113} + 12, x_{125} + 30$ |
| 22 | $x_{115}$ | Hoorn | Amsterdam CS | $x_{114} + 74$ |
| 23 | $x_{116}$ | Hoorn | Enkhuizen | $x_{118} + 52, x_{98} + 84, x_{131} + 56$ |
| 23 | $x_{117}$ | Enkhuizen | Amsterdam CS | $x_{116} + 23$ |
| 23 | $x_{118}$ | Zaandam | Amsterdam CS | $x_{117} + 50, x_{103} + 70$ |
| 24 | $x_{119}$ | Uitgeest | Amsterdam CS | $x_{121} + 29, x_{99} + 24, x_{127} + 56$ |
| 24 | $x_{120}$ | Haarlem | Alkmaar | $x_{119} + 52, x_{102} + 41$ |
| 24 | $x_{121}$ | Uitgeest | Alkmaar | $x_{120} + 23$ |
| 25 | $x_{122}$ | Uitgeest | Amsterdam CS | $x_{124} + 29, x_{30} + 50, x_{94} + 33$ |
| 25 | $x_{123}$ | Amsterdam CS | Alkmaar | $x_{122} + 38, x_{33} + 39$ |
| 25 | $x_{124}$ | Uitgeest | Alkmaar | $x_{123} + 37$ |
| 26 | $x_{125}$ | Alkmaar | Utrecht | $x_{128} + 15, x_{30} + 36$ |
| 26 | $x_{126}$ | Amsterdam CS | Utrecht | $x_{125} + 42$ |
| 26 | $x_{127}$ | Utrecht | Alkmaar | $x_{126} + 28$ |
| 26 | $x_{128}$ | Uitgeest | Alkmaar | $x_{127} + 56, x_{115} + 43$ |
| 27 | $x_{129}$ | Alkmaar | Utrecht | $x_{132} + 32, x_{36} + 36$ |
| 27 | $x_{130}$ | Zaandam | Utrecht | $x_{129} + 30, x_{121} + 45$ |
| 27 | $x_{131}$ | Utrecht | Alkmaar | $x_{130} + 40$ |
| 27 | $x_{132}$ | Zaandam | Alkmaar | $x_{131} + 29, x_{113} + 12, x_{117} + 50, x_{21} + 41$ |
| 28 | $x_{133}$ | Lelystad Centrum | Den Haag CS | $x_{136} + 36$ |
| 28 | $x_{134}$ | Duivendrecht | Den Haag CS | $x_{133} + 38, x_{31} + 43$ |
| 28 | $x_{135}$ | Den Haag CS | Lelystad Centrum | $x_{134} + 55$ |
| 28 | $x_{136}$ | Duivendrecht | Lelystad Centrum | $x_{135} + 53, x_{19} + 50$ |
| 29 | $x_{137}$ | Schiphol | Hoofddorp | $x_{139} + 84, x_{31} + 56, x_{33} + 52$ |
| 29 | $x_{138}$ | Schiphol | Lelystad Centrum | $x_{137} + 9, x_{21} + 35, x_{108} + 43$ |
| 29 | $x_{139}$ | Weesp | Lelystad Centrum | $x_{138} + 23, x_{147} + 16$ |
| 30 | $x_{140}$ | Amersfoort | Den Haag CS | $x_{143} + 13, x_{28} + 36$ |
| 30 | $x_{141}$ | Leiden | Den Haag CS | $x_{140} + 69, x_{37} + 45, x_{40} + 51$ |
| 30 | $x_{142}$ | Leiden | Amersfoort | $x_{141} + 40, x_{60} + 41$ |
| 30 | $x_{143}$ | Hilversum | Amersfoort | $x_{142} + 57, x_{147} + 32$ |
| 31 | $x_{144}$ | Weesp | Amsterdam CS | $x_{145} + 37, x_{20} + 84, x_{12} + 117, x_{11} + 114, x_{133} + 28$ |
| 31 | $x_{145}$ | Hilversum | Amersfoort | $x_{144} + 36$ |
| 32 | $x_{146}$ | Weesp | Amsterdam CS | $x_{148} + 52, x_{139} + 62$ |
| 32 | $x_{147}$ | Amsterdam CS | Utrecht | $x_{146} + 15, {}_{85} + 16$ |
| 32 | $x_{148}$ | Hilversum | Utrecht | $x_{147} + 32$ |
| 33 | $x_{149}$ | Utrecht | Hoofddorp | $x_{152} + 36, x_{50} + 29, x_{160} + 12$ |
| 33 | $x_{150}$ | Weesp | Hoofddorp | $x_{149} + 36, x_{154} + 62$ |
| 33 | $x_{151}$ | Hoofddorp | Utrecht | $x_{150} + 27, x_{52} + 19$ |
| 33 | $x_{152}$ | Weesp | Utrecht | $x_{151} + 27, x_{133} + 28, x_{153} + 31$ |
| 34 | $x_{153}$ | Weesp | Amsterdam CS | $x_{154} + 62, x_{149} + 36$ |
| 34 | $x_{154}$ | Weesp | Lelystad Centrum | $x_{153} + 31, x_{151} + 27, x_{40} + 44$ |
| 35 | $x_{155}$ | Weesp | Amsterdam CS | $x_{156} + 54, x_{140} + 24$ |
| 35 | $x_{156}$ | Weesp | Lelystad Centrum | $x_{155} + 24, x_{142} + 44$ |
| 36 | $x_{157}$ | Amsterdam CS | Rotterdam CS | $x_{159} + 41, x_{155} + 12, x_{31} + 33, x_{119} + 38$ |
| 36 | $x_{158}$ | Gouda | Rotterdam CS | $x_{157} + 51, x_{6} + 19, x_{163} + 35$ |
| 36 | $x_{159}$ | Woerden | Amsterdam CS | $x_{158} + 58, x_{160} + 24$ |
| 37 | $x_{160}$ | Woerden | Utrecht | $x_{161} + 41, x_{158} + 58, x_{108} + 53$ |
| 37 | $x_{161}$ | Alphen a/d Rijn | Leiden | $x_{160} + 37, x_{148} + 44, x_{42} + 43, x_{162} + 39$ |
| 38 | $x_{162}$ | Alphen a/d Rijn | Gouda, Alphen a/d Rijn | $x_{162} + 39, x_{161} + 26, x_{6} + 39, x_{157} + 71$ |
| 39 | $x_{163}$ | Woerden | Utrecht | $x_{164} + 54, x_{157} + 40, x_{85} + 62, x_{87} + 80, x_{107} + 65$ |
| 39 | $x_{164}$ | Woerden | Leiden | $x_{163} + 24, x_{158} + 58, x_{130} + 52$ |
| 40 | $x_{165}$ | Gouda | Alphen a/d Rijn, Gouda | $x_{165} + 39, x_{1} + 20, x_{19} + 30, x_{164} + 58$ |
| 41 | $x_{166}$ | Utrecht | Eindhoven | $x_{168} + 22, x_{42} + 28$ |
| 41 | $x_{167}$ | 's-Hertogenbosch | Eindhoven | $x_{166} + 37, x_{174} + 65$ |
| 41 | $x_{168}$ | Geldermalsen | Utrecht | $x_{167} + 76, x_{45} + 92, x_{174} + 48$ |
| 42 | $x_{169}$ | 's-Hertogenbosch | Eindhoven | $x_{171} + 76$ |
| 42 | $x_{170}$ | Eindhoven | Utrecht | $x_{169} + 30, x_{66} + 325$ |
| 42 | $x_{171}$ | 's-Hertogenbosch | Utrecht | $x_{170} + 29, x_{69} + 44$ |
| 43 | $x_{172}$ | Dordrecht | Gorinchem, Dordrecht | $x_{172} + 46, x_{58} + 57, x_{75} + 44$ |
| 44 | $x_{173}$ | Geldermalsen | Dordrecht | $x_{174} + 48, x_{171} + 17$ |
| 44 | $x_{174}$ | Dordrecht | Geldermalsen | $x_{173} + 51, x_{53} + 31, x_{56} + 22$ |
| 45 | $x_{175}$ | Amersfoort | Ede-Wageningen | $x_{176} + 34, x_{7} + 55, x_{8} + 54, x_{12} + 93, x_{11} + 90, x_{20} + 60$ |
| 45 | $x_{176}$ | Ede-Wageningen | Amersfoort | $x_{175} + 34, x_{32} + 22, x_{37} + 50$ |
| 46 | $x_{177}$ | Arnhem | Winterswijk | $x_{179} + 68, x_{32} + 33, x_{74} + 33, x_{77} + 44$ |
| 46 | $x_{178}$ | Winterswijk | Arnhem | $x_{177} + 66, x_{182} + 38$ |
| 46 | $x_{179}$ | — | — | $x_{178}$ |
| 47 | $x_{180}$ | Apeldoorn | Winterswijk | $x_{182} + 17, x_{2} + 39$ |
| 47 | $x_{181}$ | Zutphen | Winterswijk | $x_{180} + 18, x_{68} + 35, x_{197} + 39$ |
| 47 | $x_{182}$ | Zutphen | Apeldoorn | $x_{181} + 67$ |
| 59 | $x_{183}$ | Stavoren | Leeuwarden | $x_{208} + 49$ |
| 48 | $x_{184}$ | Almelo | Mariënberg | $x_{185} + 27, x_{19} + 17$ |
| 48 | $x_{185}$ | Mariënberg | Almelo | $x_{184} + 25, x_{189} + 34, x_{190} + 45$ |
| 49 | $x_{186}$ | Zwolle | Emmen | $x_{187} + 48, x_{79} + 20, x_{9} + 35, x_{12} + 57, x_{11} + 54$ |
| 49 | $x_{187}$ | Emmen | Zwolle | $x_{186} + 51$ |
| 50 | $x_{188}$ | Mariënberg | Emmen | $x_{190} + 45, x_{184} + 25, x_{22} + 88, x_{23} + 94, x_{26} + 86, x_{27} + 88$ |
| 50 | $x_{189}$ | Emmen | Zwolle | $x_{188} + 34$ |
| 50 | $x_{190}$ | Mariënberg | Zwolle | $x_{189} + 34, x_{184} + 25$ |
| 51 | $x_{191}$ | Leeuwarden | Groningen | $x_{191} + 49, x_{10} + 58, x_{201} + 41, x_{203} + 49$ |
| 51 | $x_{192}$ | Groningen | Leeuwarden | $x_{192} + 50, x_{10} + 54, x_{205} + 37$ |
| 52 | $x_{193}$ | Leeuwarden | Groningen | $x_{194} + 53, x_{10} + 58, x_{201} + 41, x_{203} + 49$ |
| 52 | $x_{194}$ | Groningen | Leeuwarden | $x_{193} + 48, x_{25} + 46, x_{207} + 37$ |
| 53 | $x_{195}$ | Hengelo | Enschede | $x_{197} + 76, x_{73} + 58, x_{74} + 49, x_{180} + 55$ |
| 53 | $x_{196}$ | Enschede | Zutphen | $x_{195} + 10$ |
| 53 | $x_{197}$ | Hengelo | Zutphen | $x_{196} + 10, x_{3} + 33$ |
| 54 | $x_{198}$ | Groningen | Roodeschool | $x_{199} + 44, x_{193} + 50$ |
| 54 | $x_{199}$ | Roodeschool | Groningen | $x_{198} + 43$ |
| 55 | $x_{200}$ | Groningen | Delfzijl | $x_{201} + 41, x_{25} + 50, x_{193} + 50$ |
| 55 | $x_{201}$ | Delfzeil | Groningen | $x_{200} + 39$ |
| 56 | $x_{202}$ | Groningen | Nieuweschans | $x_{203} + 49, x_{191} + 53, x_{10} + 58$ |
| 56 | $x_{203}$ | Nieuweschans | Groningen | $x_{202} + 49$ |
| 57 | $x_{204}$ | Leeuwarden | Harlingen Haven | $x_{205} + 37, x_{192} + 52, x_{10} + 54, x_{209} + 51$ |
| 57 | $x_{205}$ | Harlingen | Leeuwarden | $x_{204} + 31$ |
| 58 | $x_{206}$ | Leeuwarden | Harlingen Haven | $x_{207} + 37, x_{194} + 48, x_{25} + 46$ |
| 58 | $x_{207}$ | Harlingen Haven | Leeuwarden | $x_{206} + 30$ |
| 59 | $x_{208}$ | Leeuwarden | Stavoren | $x_{183} + 51, x_{205} + 37, x_{192} + 52, x_{10} + 54$ |